What is claimed is:

1. A system for generating a tree-based datastore comprising:

a processor;

a memory coupled to the processor; and

a tree-based datastore generator for creating at least one level of a tree-based datastore, the at least one level of the tree-based datastore comprising a first tree comprising a first root and at least one node of a plurality of nodes, a second tree comprising a second root and the at least one node of the first tree and at least a third tree comprising a third root and at least one of the plurality of nodes of the first tree.

2. The system of claim 1, wherein the nodes of the datastore include at least one elemental node, one subcomponent node and one end product node.

3. The system of claim 2, wherein the tree-based datastore comprises at least a first level and a second level and the end products of the first level comprise the elemental nodes of the second level.

4. The system of claim 2, wherein the tree-based datastore comprises at least a first level and a second level and the end products of the first level are decomposed to create the elemental nodes of the first level.

5. The system of claim 2, wherein the tree-based datastore comprises at least a first level and a second level and the elemental nodes of the second level are decomposed to create the elemental nodes of the first level.

6. The system of claim 1, wherein the nodes comprise pointers to other nodes in the tree-based datastore.

7. The system of claim 1, wherein the first root and second root comprise pointers to other nodes in the tree-based datastore and include non-pointer information.

8. The system of claim 7, wherein the first root is associated with a begin indicator and accesses the tree-based datastore in a first hierarchical order.

9. The system of claim 7, wherein the second root is associated with an end indicator and accesses the tree-based datastore in an inverted first hierarchical order.

10. The system of claim 7, wherein the third root includes non-pointer information associated with an element of a dataset and accesses the tree-based datastore in a third hierarchical order based on the dataset element.

11. The system of claim 1, further comprising a tree-based datastore accessor for retrieving information from the tree-based datastore.

12. The system of claim 11, wherein the tree-based datastore accessor further comprises:

means for receiving a request for information to be retrieved from the tree-based datastore;

means for retrieving the requested information from the tree-based datastore; and

means for returning the retrieved information from the tree-based datastore.

13. A system for generating a datastore comprising:

a processor;

a memory coupled to the processor; and

a datastore generator for creating a datastore, wherein the datastore comprises:

root nodes and non-root nodes organized into a plurality of connected trees, the plurality of connected trees comprising a tree of a first type comprising a first root and at least one of a plurality of non-root nodes and at least one of a plurality of trees of a second type, wherein the second tree type comprises a second root node and a plurality of non-root nodes common with the nodes of the tree of the first type.

14. The system of claim 13, wherein the nodes of the datastore include at least one elemental node, one subcomponent node and one end product node.

15. The system of claim 14, wherein the first type tree records the sequential synthesis of an end product from at least one combination of a subcomponent node with an elemental node and provides access to data in the datastore in a first context.

16. The system of claim 13, wherein at least one tree of the plurality of trees of the second type provides access to the data in the datastore in a second context.

17. The system of claim 14, wherein the at least one elemental node comprises a first null pointer, a second null pointer, a third null pointer and a fourth pointer pointing to a second list of pointers to nodes, the second list comprising nodes which contain the elemental node as their second portion.

18. The system of claim 14, wherein the at least one subcomponent node comprises a first pointer to a first node, the first node comprising a first portion of the subcomponent node, a second pointer to a second node, the second node comprising a second portion of the subcomponent node, a third pointer pointing to a first list of pointers to nodes, the first list comprising nodes which contain the subcomponent as their first portion and a fourth null pointer.

19. The system of claim 14, wherein the at least one end product node comprises at least a first pointer to a first portion, a second pointer to an ending indicator second portion, a third null pointer and a fourth pointer pointing to a second list of pointers to nodes, the second list comprising nodes which contain the end product node as their second portion.

20. The system of claim 13, wherein a root node representing a begin indicator comprises a first null pointer, a second null pointer, a third pointer pointing to a first list of pointers to nodes, the first list comprising nodes comprising the begin indicator as a first portion and a fourth null pointer.

21. The system of claim 13, wherein a root node representing a dataset element comprises a first null pointer, a second null pointer, a third null pointer and a fourth pointer pointing to a second list of pointers to nodes, the second list comprising nodes comprising the dataset element as a second portion.

22. The system of claim 13, wherein a root node representing an end product comprises a first null pointer, a second null pointer, a third null pointer and a fourth pointer pointing to a second list of pointers to nodes, the second list comprising nodes of a second level comprising the end product as a second portion.

23. The system of claim 13, further comprising an accessor for accessing information stored in the interlocking tree datastore.

24. The system of claim 23, wherein the accessor further comprises:

means for receiving a request for information to be retrieved from the tree-based datastore;

means for retrieving the requested information from the tree-based datastore; and

means for returning the retrieved information from the tree-based datastore.

25. A method for generating and accessing data from a datastore, comprising:

in response to receiving data, creating at least one level of a tree-based datastore, the at least one level of the tree-based datastore comprising a first tree comprising a first root and at least one node of a plurality of nodes, a second tree comprising a second root and the at least one node of the first tree and at least a third tree comprising a third root and at least one of the plurality of nodes of the first tree;

receiving an information request for information accessible from the at least one level of the tree-based datastore; and

retrieving the information from the at least one level of the tree-based datastore.

26. The method of claim 25, comprising:

in response to receiving data to be added to the tree-based datastore, creating a new node in the tree-based datastore for storing information associated with the received data;

creating links from the new node to a first node comprising a first portion of the new node and a second node comprising a second portion of the new node;

adding a link to the new node to a first list of pointers of the first node; and

adding a link to the new node to a second list of pointers of the second node.

27. The method of claim 26, wherein creating a new node in the tree-based datastore comprises combining a first node representing a begin indicator with a second node representing a dataset element to generate a third node representing an incomplete product.

28. The method of claim 27, wherein creating a new node in the tree-based datastore comprises combining a first node representing a first incomplete product with a second node representing a dataset element to generate a third node representing a second incomplete product.

29. The method of claim 27, wherein creating a new node in the tree-based datastore comprises combining a first incomplete product with a fourth node representing an end indicator to generate a fifth node representing an end product.

30. The method of claim 27, wherein creating a new node in the tree-based datastore comprises combining a second incomplete product with a fourth node representing an end indicator to generate a fifth node representing an end product.

31. The method of claim 26, wherein creating a new node comprises allocating space for a first pointer, a second pointer, a third pointer and a fourth pointer.

32. The method of claim 26, wherein the third pointer and fourth pointer are null pointers.

33. A method for accessing information from a tree-based datastore, the method comprising:

in response to receiving a request for information from a datastore, the information request comprising at least one constraint, the datastore comprising at least one level of a tree-based datastore, the at least one level of the tree-based datastore comprising a first tree comprising a first root and at least one node of a plurality of nodes, a second tree comprising a second root and the at least one node of the first tree and at least a third tree comprising a third root and at least one of the plurality of nodes of the first tree,

retrieving a list of nodes associated with the third root, each node in the list of nodes comprising a first portion and a second portion, wherein the third root comprises the at least one constraint; and

following the branch of each node in the first tree to the at least one node of the second tree.

34. The method of claim 33, wherein the request for information comprises a first constraint and a second constraint and the intersection of the sets of nodes of the second tree is returned.

35. A computer-readable medium comprising computer-executable instructions for generating an interlocking tree database, including instructions for:

in response to receiving data to be added to the tree-based datastore, creating a new node in the tree-based datastore for storing information associated with the received data;

creating links from the new node to a first node comprising a first portion of the new node and a second node comprising a second portion of the new node;

adding a link to the new node to a first list of pointers of the first node; and

adding a link to the new node to a second list of pointers of the second node.